DOCUMENT RESUME

ED 354 915 IR 054 501

AUTHOR	Agenbroad, James Edward
TITLE	Nonromanization: Prospects for Improving Automated Cataloging of Items in Other Writing Systems. Opinion Papers No. 3.
INSTITUTION	Library of Congress, Washington, D.C.
PUB DATE	92
NOTE	20p.; Version of a paper presented at a meeting of the Library of Congress Cataloging Forum (Washington, DC, July 22, 1991). A product of the Cataloging Forum.
AVAILABLE FROM	Cataloging Forum Steering Committee, Library of Congress, Washington, DC 20402.
PUB TYPE	Reports - Evaluative/Feasibility (142) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Bibliographic Records; Classification; Cyrillic Alphabet; Ideography; Indo European Languages; Library Automation; Library Catalogs; *Library Technical Processes; *Machine Readable Cataloging; *Non Roman Scripts; Online Catalogs; Semitic Languages; Standards; *Written Language
IDENTIFIERS	Asian Languages; Indic Languages; MARC; *Unicode

ABSTRACT

The dilemma of cataloging works in writing systems other than the roman alphabet is explored. Some characteristics of these writing system are reviewed, and the implications of these characteristics for input, retrieval, sorting, and display needed for adequate online catalogs of such works are considered. Reasons why needs have not been met are discussed, and some of the ways they might be met are examined. The following are four groups into which non-roman systems are generally divided for simplicity and features that have implications for cataloging: (1) European scripts--upper and lower case (Greek, Cyrillic, and Armenian); (2) Semitic scripts--read right to left (Hebrew and Arabic); (3) Indic scripts--implicit vowel (indigenous scripts of India and Nepal); and (4) East Asian scripts--very large character repertoires (Chinese, Korean, and Japanese). Unicode, which is an effort to define a character set that includes the letters, punctuation, and characters for all the world's writing systems offers assistance in cataloging, and will probably become an international standard late in 1992. Uses of Unicode and the MARC format are explored. (Contains 17 references.) (SLD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it
☐ Minor changes have been made to improve reproduction quality
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

IR

Opinion Papers No. 3

ED354915

NONROMANIZATION:

PROSPECTS FOR IMPROVING AUTOMATED CATALOGING OF ITEMS IN OTHER WRITING SYSTEMS

James Edward Agenbroad
Senior Systems Analyst
Information Technology Services

Cataloging Forum
Library of Congress
Washington, D.C. ■ 1992

IR 054 501

BEST COPY AVAILABLE

Opinion Papers No. 3

NONROMANIZATION:

PROSPECTS FOR IMPROVING AUTOMATED CATALOGING OF ITEMS IN OTHER WRITING SYSTEMS

James Edward Agenbroad
Senior Systems Analyst
Information Technology Services

Cataloging Forum

Library of Congress
Washington, D.C. ■ 1992

Library of Congress Cataloging-in-Publication Data

Agenbroad, James Edward, 1934-
Nonromanization : prospects for improving automated cataloging of items in other writing systems / James Edward Agenbroad.
16 p. ; 28 cm. -- (Opinion papers ; no. 3)
Includes bibliographical references.
---- ------ Copy 3 Z663 .N65 1992
1. Cataloging of foreign language publications--Data processing. 2. Transliteration--Automation. 3. Online catalog. 4. Library of Congress--Automation. I. Title. II. Series.
Z699.5.F67A36 1992
025.3' 16--dc20 92-36934
CIP

This is one in a series of occasional papers devoted to cataloging policy and practice. The opinions expressed are the author's and not necessarily those of the Cataloging Forum.

Copies of this publication are available from members of the Cataloging Forum Steering Committee.

Nonromanization: Prospects for Improving Automated Cataloging of Items in Other Writing Systems

SUMMARY

This paper describes the dilemma of cataloging works in other writing systems, outlines some characteristics of these writing systems, discusses the implications of these characteristics for input, retrieval, sorting and display needed for adequate online catalogs of such works, suggests some reasons these needs have not been met and explores some ways they might be met, e.g., Unicode.[1]

INTRODUCTION

The people of the world write and read documents in many systems other than the roman alphabet. Libraries acquire documents in nonroman scripts so readers can study and better understand these people. At LC over a third of current book cataloging is for nonroman items. To organize and service these documents librarians use romanization because the resulting records are easy to interfile with ones for roman alphabet documents—thus creating a catalog of an entire collection in a single A to Z sequence. Readers of nonroman documents, on the other hand, want to see the original script because it is more familiar to them than romanized versions of text for authors, titles etc. Few of us would recognize our names rendered in Arabic or Devanagari script (figure 1) but we routinely expect those seeking books in such scripts to use romanized versions of headings for works they want.[2]

[1] This paper is an updated version of a talk given on July 22, 1991 at a meeting of the Library of Congress' Cataloging Forum. The opinions expressed in it are purely personal, not a commitment by ITS to develop such systems.

[2] Those wanting to explore more fully the adequacy of romanization for bibliographic control should consult two articles: C. Sumner Spalding, "Romanization Reexamined," *Library Resources & Technical Services* 22, no.1 (Winter 1977): 3–12 and Hans H. Wellisch, "Multiscript and Multilingual Bibliographic Control: Alternatives to Romanization," *Library Resources & Technical Services* 22, no.2 (Spring 1978): 179–90.

Figure 1: "James Agenbroad" in Arabic and Devanagari

To accommodate the wants of librarians and readers the cataloging rules provide for giving headings in the roman alphabet, but descriptive elements in their original script whenever possible. In other words, in the card catalog era, if readers could guess how librarians romanized the heading for author or title they sought, then they could find the card with the original script which they could then read. (The need to help readers understand our romanization schemes could partially account for a need for more reference librarians in divisions that deal with these scripts than in divisions that handle only roman alphabet items.) To further assist readers for whom romanized headings are unclear, the group that approves changes to the MARC formats, the ALA Interdivisional Committee on Machine-Readable Bibliographic Information (MARBI), has added provisions in the bibliographic and authority formats respectively to allow headings and cross references from headings in other writing systems.

The LC Information Bulletin for April 13, 1979 states: "The Library reiterates that it is still firmly committed to a long-range policy of inputting machine-readable bibliographic records in a combination of nonroman and roman characters, in line with the present manual approach."

The two major bibliographic utilities, the Online Computer Library

Center (OCLC) and the Research Libraries Group (RLG), have invested considerable resources and have had commensurate success in this area. OCLC allows input, storage and display of Chinese, Japanese and Korean. RLG's Research Libraries Information Network (RLIN) handles these plus Cyrillic, Hebrew and Arabic. LC uses RLIN for cataloging books in Chinese, Japanese, Korean, Hebrew and Arabic. LC now uses OCLC for creating MARC records with the original script for Chinese, Japanese and Korean serials. Unfortunately few readers are authorized and trained to search nonroman documents on the bibliographic utilities. If they were, it would be interesting to learn their reaction to searching original script headings which the cataloging rules do not prescribe but which MARC allows. As the use of Internet and LC Direct becomes widespread readers of nonroman documents may want to search for them from a terminal in their office and then see the original script of at least the bibliographic record there.

The bulk of this paper categorizes nonroman writing systems into four groups and discusses features of each that have implications for the automation of cataloging works in each group. (Table 1) The four groups with their chief distinguishing characteristics are: European—upper/lower case; Semitic—read right to left; Indic—implicit vowel; and Han (Chinese)—very large character repertoire. (By omitting Georgian and Amharic this taxonomy and the table oversimplify the situation.)

It is important to note that just as an online catalog for items in our own alphabet requires more elaborate retrieval and sort capabilities than typical word processing software provides, an effective online catalog for items in other writing systems also requires more than the mere display of the elements of a writing system such as a Russian, Hindi or Japanese word processing program would provide. Though LC's Hebraic Section has a Hebrew script title card catalog whose sorting begins with ℵ, sorting on nonroman characters is not required by AACR2. Some users of MARC records do not have hardware needed to display nonroman writing systems. To give them some access to records containing nonroman text, the MARC format calls for also giving parallel romanized versions of all text given in other writing systems—not just the headings. Some LC romanization schemes are nearly reversible by computer programs so the feasibility of generating provisional versions of needed parallel fields will also be considered. Since several languages often use the same script while most of our romanization tables convert specific languages to our alphabet, informing the computer of

GROUP / SCRIPT	CHARACTERISTIC: Reversible Romanization	Upper/Lower Case	Known Sort Order	Initial Article	Word Space	Inflected	Direction	Context Sensitive	Diacritics
European									
Cyrillic	6	Yes	Yes	No	Yes	Yes	→	No	Yes
Greek	8	Yes	Yes	Yes	Yes	Yes	→	1 case	Yes
Armenian	7	Yes	Yes	No	Yes	Yes	→	No	No
Semitic									
Hebrew	5	No	Yes	Yes	Yes	Y/N	←	5 cases	No
Arabic	2	No	Yes	Yes	Yes	Y/N	←	Yes	Yes
South Asian									
Hindi, etc.	8	No	Yes	No	Yes	Yes	→	Yes	Yes
Tamil	8	No	Yes	No	Yes	Yes	→	No	Yes
Southeast Asian									
Burmese, Thai, Khmer and Lao	?	No	?	No	No	?	→	Yes	Yes
East Asian									
Chinese	0	No	No	No	No	No	→	No	No
Japanese	0	No	No	No	No	Yes	→	No	No
Korean	0	No	No	No	No	Yes	→	No	No

Key to Table

Reversible romanization: an estimate of how well software could derive the original script from text data romanized according to the LC scheme for a particular language; 0 = useless, 9 = accurate. *Upper/lower case:* indicates which scripts make this distinction. *Known sort order:* indicates scripts with an "alphabetic" order familiar to all its readers. *Initial article:* indicates whether the languages use articles before nouns or adjectives which need to be ignored for filing and possibly for keyword searching when they are written as part of the word as in Semitic languages and elisions, e.g., l'histoire. *Word space:* indicates whether or not the languages separate words with spaces. Those that do pose fewer problems for romanization and keyword searching. *Inflected:* indicates languages that often alter words to show grammatical categories: singular/plural, nominative/genitive, past/present, etc. *Direction:* indicates languages read from left to right or right to left. It excludes Mongolian in vertical script. *Context sensitive:* this indicates scripts whose letters vary visually depending on their environment. *Diacritics:* indicates scripts whose letters may have marks superimposed above, beneath or beside them. *Hindi, etc.:* Includes the following scripts with similar characteristics: Tibetan, Gurmukhi, Gujarati, Bengali, Oriya, Telugu, Kannada, Malayalam and Sinhalese.

Table 1: Script groups and some characteristics affecting their automation

the language of a nonroman text string would improve the performance of such software.[3]

I exclude some writing systems with minimal relevance to cataloging at LC: Mongolian in vertical script, Eskimo and Cree in Evans' syllabary, Syriac, Coptic, Cherokee, unscheduled languages of India with their own scripts, Chinese minority, i.e., non-Han, languages with their own scripts, Maldivian, traditional scripts of Indonesia and the Philippines, and extinct writing systems (deciphered or not) such as cuneiform, hieroglyphs, Indus, Easter Island, Mayan, Kharoshthi and various Central Asian scripts.

EUROPEAN SCRIPTS

This group contains scripts which distinguish between capital and lowercase letters: Greek, Cyrillic and Armenian. As it does for roman, this distinction complicates input and must be ignored during retrieval and sorting. The fewer the languages that use a script, the easier it is to define the sequence of letters for sorting. This means defining the alphabetic order of letters for Greek and Armenian presents few problems. Cyrillic script on the other hand is used not only with several Slavic languages of Europe—Russian, Serbian, Ukrainian, Bulgarian—but also, with various extra letters and diacritics, to write many Asian languages of the former Soviet Union, e.g., Uzbek. Still it is probably possible to include these special letters in the sequence of Cyrillic letters as we cope with Scandinavian letters when sorting roman letters. Greek has initial articles that must be ignored, the others do not. Greek is mildly context sensitive—one letter, sigma, appears differently at the end of a word. If the final sigma is separately keyed and stored with its own code this may need to be normalized for filing. If, instead, a single code for lower case sigma is used, the output software (printing and terminal displays) must look ahead to determine which form is wanted. Otherwise display of these scripts is not harder than doing the roman alphabet. In inflected languages words change to show number, gender, case, etc. English is slightly inflected so when using the FIND command for keyword

[3] A recent article on stemming, i.e., reducing words to their uninflected root forms, demonstrates the importance of knowing the language of the text being processed: Mirko Popovič and Peter Willett, "The Effectiveness of Stemming for Natural-Language Access to Slovene Textual Data," *JASIS, Journal of the American Society for Information Science* 43, no.5 (June 1992): 384-90.

searching one must seek the singular and plural forms of nouns. Several of these languages are quite inflected so keyword searching as implemented in MUMS would be less effective. For example, if nouns in a language have four cases (nominative, genitive, dative, and accusative), and two numbers (singular and plural), one would need to search for eight (4 x 2) forms of each noun. Writing software to generate provisional versions of romanized fields from the original script for cataloger review is probably worth exploring for Greek and the major Slavic languages that use Cyrillic—assuming the language code is present.

SEMITIC SCRIPTS

This group contains Hebrew and Arabic scripts. Hebrew is used with a few other languages, mainly Yiddish; Arabic is used with many languages including Persian, Urdu, Pushto, Tajik, Sindhi, Kashmiri, Uighur and Malay. As with the roman and Cyrillic scripts, there are extra letters and diacritics for languages other than Arabic. Not just titles but also Hebrew and Arabic personal names have initial articles which must be ignored in sorting. Articles are not written separately (like the French word "l'histoire") which makes keyword retrieval more difficult. Many vowels are seldom written and should probably be ignored for sorting. Current LC romanization schemes call for supplying the vowels which is quite labor intensive. This means generation of provisional parallel fields for catalogers to review could probably only generate the original script from the romanized form (rather than vice versa) since the computer could not predict the vowels. For automation the chief difficulty is that these languages are written and read from right to left. This poses major problems for transmission, sorting and display. Though it appears at the right margin, the first letter of an Arabic or Hebrew title is wanted first in 245 field of the MARC record so an effective title key (PTK) can be built. This is also important for sorting. Unlike letters, numbers are written and read in the same direction as they are in roman titles; this complicates keying, transmitting, storing, sorting and displaying a Semitic title similar to "76 trombones". The need to combine in a single field Semitic and left to right text strings (e.g., the title of a Hebrew/Russian dictionary) makes matters even more difficult. Like Greek, Hebrew is slightly context sensitive—five letters have a separate final form to be dealt with.

Arabic is very context sensitive—all but a few letters appear in four forms depending on their position in a word—initial, middle, final or with a

space on both sides of it. For example the letter Ba alone is ب ; at the end, middle and start of a word it appears as ـب بـ بـ respectively. In most modern Arabic text computer systems (including RLG's) one keys a single letter (regardless of its position) which is stored with a single code. Then the display software determines and generates the appropriate visual form. Many special letter combinations analogous to roman ligatures such as fi and ffl are desirable for high class Arabic typography, but it is mandatory to use the lam-alif combination whenever these letters occur together. If, however, this combination is stored with its own code, sorting software must expand it. While Urdu uses the Arabic script, instead of a linear right to left sequence it uses the nastaliq style in which words and phrases usually appear diagonally e.g., ✓3 ✓2 ✓1 نستعلیق . When cards for Urdu items displayed Urdu they used horizontal Arabic type, not nastaliq, so perhaps the online catalog need not do so either. At least one Central Asian country formerly part of the Soviet Union, Tajikistan, again allows printing in an expanded version of Arabic.

INDIC SCRIPTS

By Indic scripts I mean the indigenous scripts of India and Nepal: Devanagari (for Hindi, Marathi and Nepali), Gurmukhi (for Panjabi), Gujarati, Bengali, Oriya, Telugu, Kannada, Tamil and Malayalam, and the related scripts used in Tibet, Sri Lanka and Southeast Asia (Burmese, Thai, Lao, Khmer and Javanese in Kawi script).

My knowledge is largely limited to the scripts used in India. While these scripts look very different, in almost all cases they share the following characteristics: 1. Alphabetic order—the vowels come first followed by consonants from K produced at the back of the throat to M produced with the lips. 2. The most common vowel sound "a" is implicit in consonants, not written unless it begins a syllable. 3. Except at the start of syllables, other vowels are written as modifiers of consonants—above, below, on one or both sides of the consonant—where they override the implicit consonant. 4. When a consonant has no vowel because it is pronounced together with one or more following consonants (e.g., "st") the consonants are written in a fused form called a conjunct consonant. For example, in Devanagari script which is in all probability the most widely used alphabet of South Asian origin, Sa = स and Ta = त but Sta = स्त . (Figure 2 shows vowel

BEST COPY AVAILABLE

modifiers and conjunct consonants for the word "moonlight" in Hindi, Tamil and Malayalam.)

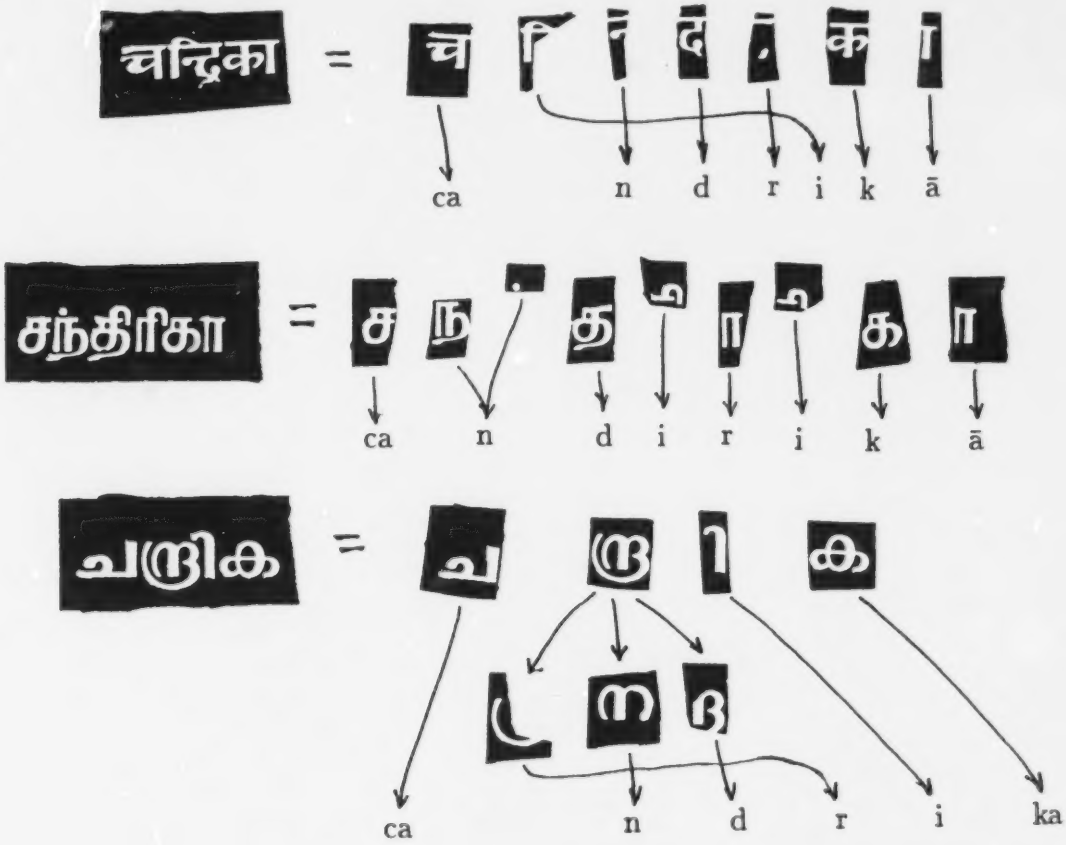

Figure 2: The word "candrika" in Hindi, Tamil and Malayalam

In India, though words can be quite long, they are written with spaces between them. In Southeast Asian scripts spaces do not separate words. Keyword extraction and retrieval will be difficult for languages that do not use spaces. Some of the languages using these scripts are highly inflected; like the need to request both the singular and plural forms of English nouns with the FIND command, these inflected forms make keyword retrieval more difficult. Keying is not particularly difficult. So long as there is a means to indicate the absence of a vowel, display programs, though complex, can be

devised to cope with vowel indicators and conjunct consonants—the Indians have written software to do so. The order of these alphabets presents few problems for sorting programs. Certain consonantal sounds that follow vowels but by Indian filing tradition cause a syllable to precede the same letters without the following consonant may prove difficult. The romanization schemes LC uses are sufficiently reversible to make generation of provisional versions of romanized text worth exploring—at least for languages that use spaces.

EAST ASIAN SCRIPTS

Unlike the previously discussed writing systems which use fewer than a hundred "letters" assigned to components of the sound system of a particular language, Chinese is written with thousands of different characters which more nearly represent either the idea of a word or its idea and its sound. Japanese uses these characters (calling them kanji) and about forty other characters (called kana) that represent sounds much as the roman alphabet does. Similarly, South Koreans write with a mixture of Chinese (called hanji) and syllabic characters (called hangul). Hangul syllables are built from separate elements for the constituent vowel and consonants which is somewhat analogous to building syllables in Indic scripts. In North Korea only hangul are used. For purposes of automation Japanese kana and Korean hangul pose no new difficulties—they are few in number and have a known sequence for sorting.[4] All three languages are written without spaces which makes keyword indexing and retrieval difficult. The existence of traditional and simplified forms of many Chinese characters which must be displayed differently but treated as the same for retrieval and sorting purposes further complicates matters. Procedures for assigning word boundaries for romanized texts are complex and time consuming. Keyword access will be ineffective unless a searcher's notion of what constitutes a word matches the cataloger's.

It is the sheer number of their characters that makes these writing systems challenging both to readers and computers. There are far too many

[4] I recently learned that North and South Korea use different sort sequences for hangul but there is a proposal to unify them. For details see Kyongsok Kim, "A Future Direction in Standardizing International Character Codes—with Special Reference to ISO/IEC 10646 and Unicode," *Computer Standards & Interfaces* 14, no.3 (May 1992): 209–21.

to fit onto a single keyboard so various input schemes exist. Typically they involve keying an approximation of a character—its shape, its sound, its strokes or some combination of them—and then selecting the desired character from a menu of those that match the approximation. Because there are so many characters, there is no one widely accepted collating sequence for them analogous to our A-Z alphabetic order. Instead, there are many different schemes for sequencing Chinese characters. The Japanese and Koreans generally sort their characters by the accepted order of their sounds as represented in kana and hangul respectively. It would be possible to store the kana and hangul equivalents for sorting. For Chinese and Korean, generation of provisional romanized equivalents might be possible. For Japanese, doing so is less promising because many kanji have two pronunciations. Because Chinese characters are very intricate their display at terminals or printers requires higher resolution devices which also cost more. Their number and higher resolution requirements mean more storage. While the number of Chinese characters is finite, it is not fixed so a method is needed to add characters occasionally to the input and output devices.

CONCLUSION

This paper has not listed every detail of every writing system found in works LC catalogs. A few other factors must be mentioned. For reasons of widest possible utility the MARC format is by intention independent of a single hardware or software vendor's offerings. This has consequences for costs and speed of development. If LC could go it alone we would be further than we are. Second, while LC acquires many materials written in nonroman scripts, their users are far from a united and vocal audience. If they were we would have made more effort to satisfy their needs. Elsewhere work has been done with automation of virtually every script mentioned (and even some of those I excluded). Until recently this work has usually involved roman and one other writing system; on the other hand, continuing the Library's integrated catalog requires a many scripts aproach.

Fortunately the prospect of global markets has made the computer industry broaden its perspective. We now have on the horizon the beginnings of an all-scripts approach which comes closer to the Library's needs. This has resulted in the Unicode and the ISO 10646 efforts to define an integrated character set standard for all writing systems. If terminal and printer vendors implement this character set and if MARBI and LC adopt it too, we

could make our online catalog as legible and effective for readers as the card catalog was for finding works in other scripts. It could be even more effective if we create headings in the original script.

In the following pages (not part of my talk) I discuss some ways we could use Unicode in MARC to let us realize such improvements. We should be able to select and and implement an approach that would free us from the input, storage and display aspects of nonroman scripts so we can concentrate on the nonroman retrieval and sorting issues. The basic problem will soon be political, not technical—given our limited resources, what priority does effective catalog access and display for works in nonroman scripts have? Can those who want improvements in access to materials in other scripts raise their priority?

POSSIBLE ROLE OF UNICODE IN MARC

The preceding pages have briefly described features of various nonroman writing systems that must be dealt with to improve the cataloging functions for works in these writing systems. After a short explanation of MARC and Unicode, this section examines some ways MARC might use Unicode.

A well known byproduct of the excellence needed in a catalog of a collection as vast as the Library's has been the acceptance of LC cataloging by other libraries. Since the late sixties the medium of distribution for this cataloging has increasingly been the MARC format. This format defines a record structure and the means for identifying the elements of a bibliographic record so others can use the data for their needs. This format also includes a character set, "the ALA character set," which was revolutionary when it was introduced because it specified codes for many special characters (e.g., Æ, Ł and £) and diacritics (e.g., ẍ, ẍ, x̣, x̧, x͡x, x̃, etc., all shown here with x) needed to transcribe accurately titles in foreign languages that use our alphabet. (A character set is a repertoire of letters, punctuation, numerals, diacritics, etc. and the unique computer code assigned to each.) More recently character sets for the Cyrillic, Hebrew, Arabic alphabets and one for Chinese, Japanese and Korean characters have been added to the MARC format definition but these characters have not been implemented on systems maintained at LC. Besides the reasons already mentioned these character sets have not been implemented because it would be expensive to do so.

Unicode is an effort to define a character set that includes the letters, characters, punctuation, etc. for all the world's text writing systems. (It does not cover pictorial matter, color or musical notation but cataloging does not require them.) Unicode will probably become an international standard, ISO 10646, late in 1992. Software and terminal vendors will then begin to implement it in their products to facilitate sales to foreign and multinational customers who need to communicate widely. I expect that Unicode will be as revolutionary as the "ALA character set" once was. When terminals with Unicode become commercially available they will reduce the cost of implementing the improvements described above—but only *if* MARC adopts Unicode.

Three features of Unicode must be kept in mind. First, at present it does not contain a few characters in the ALA set, mainly the ligature used

in romanization, e.g., ts, and the double width tilde, e.g., n͡g, which is used very seldom. This could be solved either by getting them restored to Unicode (they were in a draft) or by adding them in the private use space. The former is preferable. Second, it uses 16 bits per character instead of 8. (This is how it gets enough different codes for so many characters.) The approach is roughly analogous to changing braille from six dots to twelve. It is a major change for anyone who will use Unicode. Third, the code for a diacritic follows the letter it modifies; in MARC the diacritic comes first. This is a significant change but it effects only MARC software, not all users of Unicode. It could be overcome by a database upgrade that reversed the sequence of all diacritics and changed any software that processed diacritics—not just software for input and display but for retrieval and sorting as well. Such a conversion would require close coordination with users of MARC data.

The present treatment of Indic scripts in Unicode leaves something to be desired. The codes for many letters differ from those in the relevant Indian standard, IS 13194 1991, and they should not. Some Indic scripts display some vowel signs on two sides of a consonant. Unicode has added an extra code for the second part of such signs. These are superfluous; they obscure the shared symmetry that is the hallmark of Indian scripts; unless removed they will complicate exchanging software and data with Indian organizations that follow their standard for their scripts.

Assuming the above are resolved, the Unicode options I can see are:

1. Do nothing. This would be appropriate if vendors do not implement Unicode. If they do, this would unnecessarily perpetuate and increase the separation between bibliographic and other text data processing applications. It is contrary to the trend toward networking.

2. Define Unicode as the new MARC character set so every character is 16 bits long. This would render virtually all MARC software obsolete. This is as extreme as the first option but in the opposite direction.

3. Use an escape sequence to invoke Unicode as needed. An escape sequence announces that a new character set is in effect. This is the technique now used in MARC to invoke Cyrillic, Arabic and other character sets. Though ISO has defined an escape sequence for Unicode, registration number 162, vendor implementations of Unicode may not allow this approach. A Unicode escape sequence could be adopted by MARC in at least three ways:

a. As the only escape sequence; it would be used whenever the need arose for a character not in ASCII, the US standard set which assigns codes for A–Z, a–z, 0–9 and punctuation. Most microcomputers and word processors already use ASCII.

b. As the only escape sequence; but use it only when one needed to exceed the ALA character set. Unless the diacritics conversion described above were done this would result in records with some diacritics after the letter they modified (ALA) and others before the letter they modify (Unicode). This is undesirable even if the Unicode data were always in the 880 field where all nonroman data (Cyrillic, Hebrew, etc.) now resides.

c. Use the Unicode escape as just one more escape when one needed to invoke a character set other than those now in use (i.e., Cyrillic, Arabic, Hebrew and CJK). Then Unicode would be used just for Greek, Indic and other writing systems that MARC does not now allow. This would minimize both the economic and networking advantages of using Unicode.

4. Define fields that would use Unicode exclusively. In these fields each character would be two bytes long including the indicators, delimiters, subfield codes and end of field character. Rather than define new fields, one could declare that for Unicode data the first character of each tag was alphabetic so 0=A, 1=B, etc. Then C45 (or possibly c45) would be the tag of a title field containing Unicode. While this too would result in records with diacritics before and after the letter they modify in different fields, the tag would give an early warning.

5. Dual mode distribution could also be considered. Records for which both the ALA and Unicode sets were adequate could be made available with either the ALA or Unicode character sets at the recipient's option. This assumes that Unicode would assign codes to every element of the ALA set. It could complicate networking since two versions of many records would exist.

In deciding how MARC will respond to Unicode we must weigh improved service and reduced dependence on expensive customized devices against the cost of conversion. Other factors include the risks that inaction would further isolate libraries from readers and that a subscriber might convert MARC records to Unicode and market them.

FURTHER READING

This paper covers a very broad topic. The following items may help those wanting to read more about the use of computers with other writing systems; they are in chronological order. The list does not pretend to be a comprehensive bibliography of the topic.

Languages of the World That Can Be Set on 'Monotype' Machines. Compiled by R.A. Downie. London: 1963. (The Monotype Recorder, v. 42, no.4) Good on the variety of scripts, nothing on their automation.

Om Vikas. *Use of Non-English Languages in Computers: A Selected Bibliography.* New Delhi: Electronics Commission Information, Planning & Analysis Group, 1978. Impressive with 369 entries though some pertain to other roman alphabet languages.

Akira Nakanishi. *Writing Systems of the World.* Rutland, Vt.: Tuttle, 1980. Similar to the first item.

CALTIS. Pune, India: 1983-85. Papers from three meetings on calligraphy, lettering and typography of Indian scripts.

Computer Processing of Chinese & Oriental Languages: An International Journal of the Chinese Language Computer Society. Montreal: 1983-

Joseph D. Becker. "Multilingual Word Processing." *Scientific American* 215, no.1 (July 1984): 96-109. An excellent introduction.

SESAME Bulletin: Language Automation Worldwide. Harrogate, Eng.: 1986- A quarterly journal; SESAME stands for Southeast, South Asia, Middle East.

Automated Systems for Access to Multilingual and Multiscript Library Materials: Problems and Solutions. Edited by Christine Boßmeyer and Stephen W. Massil. München, New York: K.G. Saur, 1987. (IFLA publications, 38) Papers from an IFLA pre-conference, Tokyo, August 21-22, 1986.

John Clews. *Language Automation Worldwide: The Development of Character Set Standards.* Harrogate: SESAME Computer Projects, 1988. Good on library and other character sets.

Jack K.T. Huang and Timothy D. Huang. *Introduction to Chinese, Japanese and Korean Computing.* Singapore; Teaneck, N.J.: World Scientific, 1989.

Computers and the Arabic Language. Edited by Pierre Mackay. New York: Hemisphere, 1990.

Randall K. Barry. "The Standards Dilemma of Character Sets." *Information Standards Quarterly* 3, no.2 (April 1991): 8–16. On library and other character set standards.

Kenneth M. Sheldon. "ACSII Goes Global." *Byte* 17, no.7 (July 1991): 108–15. On the two attempts to standardize computer codes for all writing systems—Unicode and ISO 10646.

Unicode Consortium. *The Unicode Standard: A Worldwide Character Encoding.* Version 1.0. Reading, Mass.: Addison-Wesley, c1991– Volume one covers all modern scripts except those for China, Japan and Korea which will appear in volume two.

Indian Script Code for Information Interchange. New Delhi: Bureau of Indian Standards, 1991. (IS 13194)

Information Technology, Universal Multiple-Octet Coded Character Set, UCS, Part 1: Architecture and Basic Multilingual Plane. (26 Dec. 1991). "Working document for ISO/IEC draft international standard 10646-1.2"

Joan M. Aliprand. "Nonroman Scripts in the Bibliographic Environment." *Information Technology and Libraries* 11, no.2 (June 1992): 105–19. Ably covers much the same ground but aimed more toward systems people. Discusses ways MARC might incorporate a global character set such as Unicode.